

Torque control of PMSM motors using reinforcement learning agent algorithm for electric vehicle application

Vo Thanh Ha¹, Duong Anh Tuan², Tran Thuy Van²

¹Department of Cybernetics, Faculty of Electrical and Electronic Engineering, University of Transport and Communications, Hanoi, Vietnam

²Faculty of Electrical Engineering, Hanoi University of Industry, Hanoi, Vietnam

Article Info

Article history:

Received Nov 8, 2023

Revised Apr 13, 2025

Accepted May 27, 2025

Keywords:

Electric vehicles

Field-oriented control

Interior-mounted permanent

magnet synchronous motors

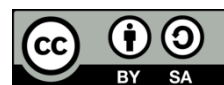
Reinforcement learning

Reinforcement learning agent

ABSTRACT

As electric vehicles (EVs) demand higher performance and efficiency, precise torque control in interior permanent magnet synchronous motors (IPMSMs) becomes increasingly vital. This paper introduces a reinforcement learning (RL)-based method to optimize torque control in IPMSMs. The RL agent is trained to regulate d-axis and q-axis currents, producing stator voltages to follow the desired motor speed. The control system includes an observation vector, voltage-based actions, and a specially designed reward function. Due to the nonlinear dynamics of the motor, training the agent requires significant computational effort. MATLAB/Simulink simulations are performed to compare the RL controller with a traditional PI controller. Results indicate that the RL controller delivers quicker and more accurate performance, although additional training is necessary to minimize overshoot.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Vo Thanh Ha

Faculty of Electrical and Electronic Engineering, University of Transport and Communications

Room 505, Building A6, No.3 Cau Giay Street, Lang Thuong Ward, Dong Da District, Hanoi, Vietnam

Email: vothanhha.ktd@utc.edu.vn

1. INTRODUCTION

Electric vehicles (EVs) require motors that not only deliver high-speed operation and compact design but also maintain a high-power density per unit volume. Additionally, the motor must exhibit mechanical robustness and high efficiency across a wide range of speeds. Therefore, selecting a suitable motor type for EV drive systems that balances multiple performance criteria is critical [1]–[3]. Among various options, the permanent magnet synchronous motor (PMSM) has become a standard choice for EVs due to its superior performance characteristics [4]–[6]. PMSMs are commonly classified into two types: surface-mounted PMSMs (SPMSMs) and interior permanent magnet synchronous motors (IPMSMs). The IPMSM stands out by leveraging both magnetic and reluctance torque, enabled by rotor saliency, which enhances torque output and broadens the operating speed range compared to SPMSM [7]. Furthermore, IPMSMs offer a wider constant-power range due to their high torque-per-ampere ratio and efficient field-weakening capability [8]. The mechanical integrity of IPMSMs is also notable, as the embedded magnets provide better structural strength [9]. To further enhance torque and efficiency, two primary approaches are applied in electric drive systems. The first involves mechanical enhancements, such as increasing the number of pole pairs in the stator to improve torque production [10], [11]. The second approach centers on the development of advanced control strategies that optimize dynamic response and energy utilization. A range of control methods has been proposed for PMSM drives. Huang *et al.* [12] provided a comprehensive review of classical and nonlinear control strategies, including field-oriented control (FOC), direct torque control

(DTC), fuzzy logic control (FLC), model predictive control (MPC), and sliding mode control (SMC), all of which improve torque tracking and robustness. Nicola and Nicola [13] implemented FOC on an FPGA platform, integrating PI controllers with space vector PWM (SVPWM), achieving high efficiency and rapid dynamic response. Demir and Vural [14] employed SMC in a PMSM traction drive powered by a multi-level inverter, which resulted in a significant reduction in torque ripple and improved system stability. Moreover, recent advancements have introduced intelligent control algorithms such as genetic algorithms (GA), particle swarm optimization (PSO), FLC, and neural networks (NN) [15], [16]. These methods optimize control behavior through learning and adaptation, especially under varying conditions. Hybrid approaches—such as combining FLC with NN, PID, or SMC—have also been explored to enhance robustness and adaptability [17], [18]. While nonlinear controllers like MPC and SMC deliver strong performance, they typically require accurate system models and involve complex design procedures, particularly for systems with high-order dynamics [19], [20]. Nevertheless, they remain highly responsive and robust. In contrast, intelligent controllers such as GA, PSO, FLC, and NN primarily rely on the input error vector and its derivative, using rule-based logic to generate control actions. Although they may require longer computational time, these methods offer high accuracy and are less dependent on detailed system modelling [21]. Their fast response, precision, and reliability make them well-suited for applications involving linear systems, such as current or torque control, or systems exposed to parameter variations and environmental disturbances.

This research introduces a reinforcement learning (RL)-based strategy for controlling the torque of IPMSM motors in EV applications. The proposed reinforcement learning agent (RLAgent) adaptively modulates the stator voltages by controlling the d-axis and q-axis currents to achieve the target rotational speed. The RLAgent structure comprises three essential elements: i) the observation vector combined with the outer-loop speed reference, ii) the voltage control actions generated by the agent, and iii) the reward signal, which is dynamically calculated at each time step using a cost function. Due to its data-driven nature, the training process is computationally intensive and may require a considerable amount of time to converge [22]–[25].

The paper is divided into five sections. The first introduces the motivation and context of IPMSM motor control in EVs. The second section presents the mathematical modeling of the IPMSM system. Section 3 details the design and development of the RLAgent-based torque controller using the established model. Section 4 evaluates the proposed method through MATLAB/Simulink simulations. Finally, the paper concludes with a summary of findings, control performance analysis, and potential directions for future improvements.

2. MATHEMATICAL MODEL OF A PERMANENT MAGNET SYNCHRONOUS MOTOR AND LOAD

2.1. Mathematical model of the permanent magnet synchronous motor

The IPMSM motor is controlled using the FOC strategy, which is advantageous due to its cascaded loop structure. In the dq rotating reference frame, the mathematical equations governing the motor behavior can be expressed as (1):

$$\begin{cases} u_{sd} = L_{sd} \frac{di_{sd}}{dt} + R_s i_{sd} - \omega L_{sq} i_{sq} \\ u_{sq} = L_{sq} \frac{di_{sq}}{dt} + R_s i_{sq} + \omega L_{sd} i_{sd} + \omega \psi \end{cases} \quad (1)$$

where: i_{sd}, i_{sq} are the d and q -axis stator currents; u_{sd}, u_{sq} are the corresponding stator voltages; L_{sd}, L_{sq} are the d - and q -axis inductances; R_s is the stator resistance; ω is the electrical angular speed; and ψ is the rotor flux linkage. The electromagnetic torque of the IPMSM can be calculated using:

$$m_M = \frac{3}{2} P_c [\psi_p i_{sq} + i_{sd} i_{sq} (L_{sd} - L_{sq})] \quad (2)$$

In the case of optimal rotor flux orientation, the d -axis current is typically set to zero ($i_{sd} = 0$), simplifying the torque equation to:

$$m_M = \frac{3}{2} P_c \psi_p i_{sq} \quad (3)$$

2.2. Mathematical model of the electric vehicle drivetrain

The EV dynamics are coupled with the motor through the drivetrain. The relationship between motor torque and wheel torque is:

$$\begin{cases} T \cdot k_{gear} = T_{Wh} \\ \omega_{Wh} = \omega_m k_{gear} \end{cases} \quad (4)$$

where: T is the motor torque; torque acting on the wheel; k_{gear} is the gear ratio; ω_m is the motor angular speed; and T_{Wh} is the torque applied to the wheel.

Applying Newton's second law for rotational systems gives:

$$T - T_{Wh} = J \frac{d\omega_m}{dt} \quad (5)$$

Additionally, the motion of the wheel is described by:

$$\begin{aligned} v_{Wh} &= \omega_{Wh} R_{Wh} \\ R_{Wh} T_{Wh} &= T_L = R_{Wh} F_t \end{aligned} \quad (6)$$

where: R_{Wh} is the radius and F_t is the traction (or drag) force applied to the wheel.

3. DESIGN OF REINFORCEMENT LEARNING AGENT TORQUE CONTROLLER

RL is a machine learning approach in which an agent learns to make decisions through interaction with an environment, aiming to maximize a cumulative reward. In the proposed method, an RL agent controls the torque of an IPMSM by adjusting the stator voltages that affect the d- and q-axis currents. The way of learning to improve erection is shown in Figure 1.

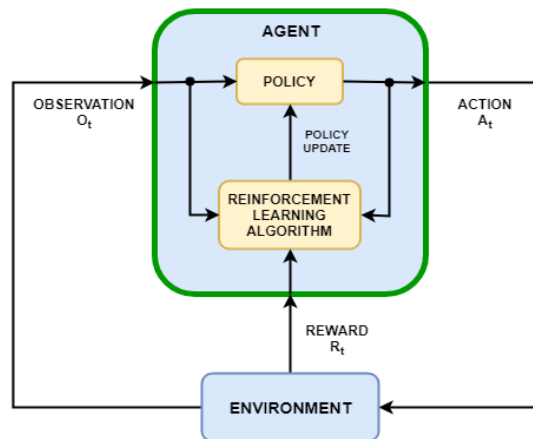


Figure 1. The structure of the RL Agent method

Figure 1 shows a RL system with an agent and an environment. The agent learns and decides, while the environment represents the system or task, like an RV. At each step t , the environment provides a state observation O_t . Using this, the agent chooses an action A_t based on how the policy affects it. The environment responds with a new observation and reward R_t , reflecting action quality. The agent adjusts its policy to optimise cumulative reward through this iterative process, improving control through experience. The objective is to find an optimal policy $\pi(s)$ that selects the best action in each state. The expected return is expressed as:

$$\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \quad (7)$$

In RL, the agent's objective is to maximize the cumulative reward over time, which is mathematically expressed as (7). This equation $r(s_t, a_t)$ denotes the immediate reward received after acting, a_t in state s_t at time step t , and γ is the discount factor determining the importance of future rewards relative to immediate ones. A discount factor $0 < \gamma < 1$ ensures that rewards received shortly are given more weight than future rewards. This formulation allows the agent to learn a policy that balances short-term gains with long-term benefits, which is essential for achieving optimal performance in dynamic environments such as EV motor control systems.

3.1. The Markov decision process

A Markov decision process (MDP) is a foundational framework in RL used to describe environments where outcomes are partly random and partly controlled by a decision-maker. It is defined by five key components: a finite set of states S , a set of possible actions A , a transition probability function P , a reward function R , and a discount factor γ . At each time step t , the agent observes the current state S_t , selects an action A_t , receives a reward $R(S_t, A_t)$, and transitions to a new state S_{t+1} according to the probability distribution $P(S_{t+1}|S_t, A_t)$. The agent's goal is to learn an optimal policy $\pi(s)$ which defines the best action to take in each state to maximise the expected cumulative reward over time. This objective is mathematically expressed as $\sum_0^\infty \gamma^t r_{a_t}(S_t, S_{t+1})$; where γ (0,1) is the discount factor that emphasises the importance of immediate rewards while gradually reducing the weight of future rewards. The optimal policy maximises this discounted return, enabling the agent to make intelligent, long-term decisions in dynamic environments.

3.2. Q-learning

In the RL method, the Q-value matrix also known as the action-value function is computed using (8), which enables the agent to evaluate the potential outcomes of different actions in each state. By updating the Q-values over time, the agent learns to estimate which action yields the highest expected cumulative reward. This process allows the agent to make optimal decisions by selecting actions that maximize the long-term return based on its current knowledge of the environment.

$$Q(s, a) = r(s, a) + \gamma \max_a Q(s', a) \quad (8)$$

In (8) represents the update rule for the Q-table, where the value of each state-action pair $Q(s, a)$ is incrementally adjusted based on the reward received $r(s, a)$ and the estimated future rewards from the next state $r(s, a)$. This update creates an action-value matrix that guides the agent's decision-making by allowing it to select the action with the highest Q-value in any given state. Since RL is inherently stochastic, the values in the Q-table evolve as the agent experiences different transitions. To accommodate this variability, the Q-values are refined using the temporal-difference (TD) error, as shown in (9). This error measures the difference between the predicted and actual rewards and ensures that the agent continually improves its policy based on new experiences.

$$Q_t(s, a) = Q_{t-1}(s, a) + \alpha TD_t(a, s) \quad (9)$$

Is calculated by for (10):

$$TD_t = R(s, a) + \gamma \max_{a'} Q(s', a') - Q_{t-1}(s, a) \quad (10)$$

where α is the learning rate factor.

3.3. Created environment for reinforcement learning agent

The training environment designed for the RLAgent consists of several critical components that define the agent's perception and interaction with the system. The observations provided to the agent include the outer-loop reference speed, the d -axis and q -axis current errors (i_d , i_q), their respective error derivatives, and the integral of those errors (i_{derror} , i_{qerror}).

These inputs allow the agent to accurately estimate the current state of the system. Based on this information, the agent outputs continuous control voltages u_d and u_q which directly affect the stator's dynamic behavior. The agent operates at a high sampling rate (measured in seconds), whereas the outer control loop typically functions at a lower frequency. The simulation runs for a maximum of 1000 time steps, unless an early termination condition is satisfied—such as reaching the desired reference value of i_q . The reward function, as defined in (11), penalizes both large current tracking errors and excessive control efforts, thereby guiding the agent to generate precise and energy-efficient control actions.

$$r_1 = - \left(Q_1 * i_{derror}^2 + Q_2 * i_{qerror}^2 + R * \sum_i u_{i_{t-1}}^2 \right) - 100d \quad (11)$$

where $Q_1=Q_2=4.5$ and $R=0.15$ are constants, i_{derror} the d -axis is the current error i_{qerror} , the current q -axis mistake is the previous time step actions, and d is a flag that equals 1 when the simulation is ended prematurely.

The environment created for the RLAgent is programmed according to the algorithmic flowchart shown in Figure 2. To learn a parameterized policy, a deterministic actor network determines optimal actions in a continuous action space. This actor takes the current observation as input and outputs a deterministic control action. The policy is modelled using a NN with a single input layer. Training is conducted using random mini batches of size 600 drawn from an experience replay buffer with a 2×10^5 capacity. To encourage long-term optimal behaviour, a discount factor of 0.995 is employed. The training architecture follows the twin delayed deep deterministic policy gradient (TD3) method, where both actor and critic target networks are updated every 10 steps using a soft update mechanism with a smoothing coefficient of 0.005.

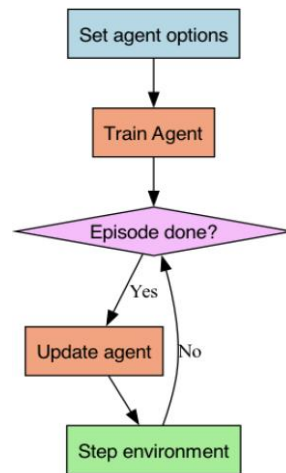


Figure 2. The algorithmic flowchart to create an environment for RLAgent

Figure 2 presents the workflow for training a RLAgent. The process begins with setting the agent's options, including learning rate, discount factor, buffer size, and policy structure. Once configured, the training phase is initiated. During each training episode, the agent interacts with the environment by executing actions based on its current policy. After each action, the environment returns feedback through new observations and rewards. The system then checks whether the episode has concluded. If the episode is complete, the agent updates its policy using the accumulated experience. Otherwise, it continues to step through the environment. This interaction and policy refinement loop continues iteratively, allowing the agent to learn and improve its decision-making over time based on the defined reward structure.

4. RESULTS SIMULATION

The simulation framework for the proposed RLAgent controller is constructed as depicted in Figure 3. This study evaluates the system under varying reference speeds of 200 rpm, 400 rpm, 600 rpm, 800 rpm, and 1000 rpm to assess its adaptability and robustness. Both the conventional PI and RL-based controllers are tested over a 10-second simulation period, during which they respond to changes in the reference speed. The RLAgent is trained using a NN model. Each training episode comprises 150-time steps, which are repeated for up to 1000 episodes. The learning rate is set to 0.003. Training is terminated when the agent achieves an average cumulative reward exceeding -190 over a rolling window of 100 episodes, indicating acceptable performance. Once trained, the agent is capable of effectively tracking various reference speeds. The technical specifications of the IPMSM motor used in the simulation are summarized in Table 1.

Table 1. The parameter for the IMSM motor

Motor parameters	Value symbol
Power	35 kW
Rated speed	3000 rpm
Rated voltage	300 V
Number of pole pairs	2
Magnetic flux density	0.0437
Maximum torque	105 Nm
Armature resistance	0.3 Ω
Shaft inductance d	8.76e-5 H
Shaft inductance q	7.72e-5 H

Figure 3 presents the integrated control architecture of an electric drive system employing a reinforcement learning-based torque controller (RLAgent) for an IPMSM. The system begins with a speed control loop, where the reference speed ω^* is compared to the rotor speed ω . A PI controller processes the resulting error to generate the reference torque T^* . Subsequently, this torque command is fed into the torque and flux estimation model, which computes the reference currents i_d^*, i_q^* where typically $i_d^* = 0$ to ensure maximum torque per ampere. These reference currents are then utilized by the RLAgent, which acts as the central torque controller. Unlike conventional controllers, the RLAgent leverages learning-based decision-making to output the voltage u_d, u_q , based on observed current errors and motor states. These voltage signals are transformed via the inverse Park transformation and passed into the space vector modulation (SVM) unit, which generates the necessary PWM signals to control the three-phase inverter. As a result, the inverter produces appropriate phase voltages to drive the IPMSM. Meanwhile, current feedback is collected through ADCs and transformed from the three-phase abc frame to the dq frame using Clarke and Park transformations, thus closing the feedback loop. Overall, this architecture seamlessly integrates traditional control with RL, enabling the system to adaptively manage torque and speed in real-time, even in the presence of nonlinearities and uncertainties typical of EV operations. The dynamic responses of the RLAgent are presented in Figures 4-6.

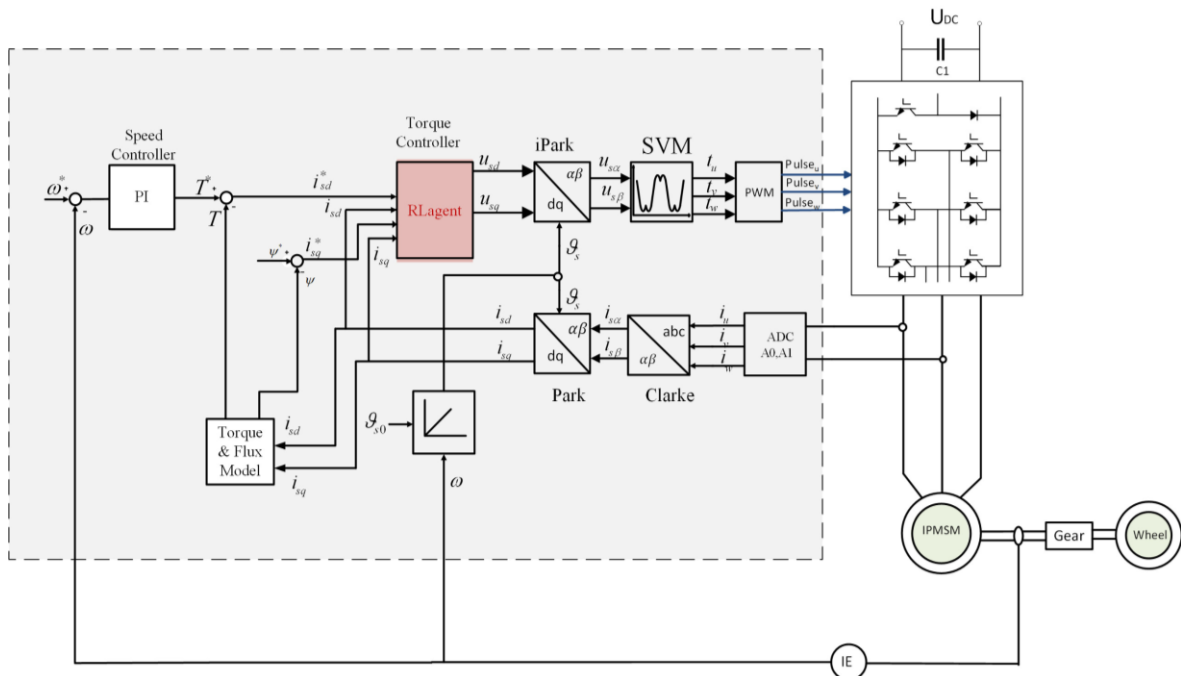


Figure 3. The simulation structure of RLAgent controller

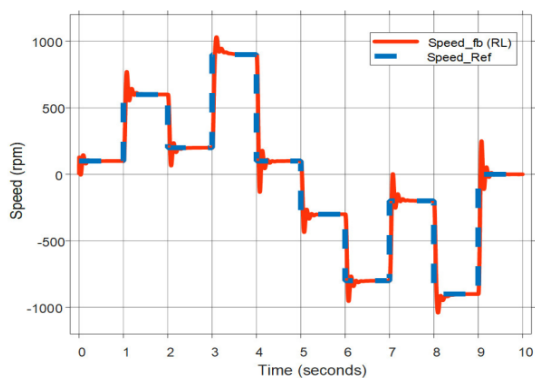


Figure 4. The speed response of the RLAgent

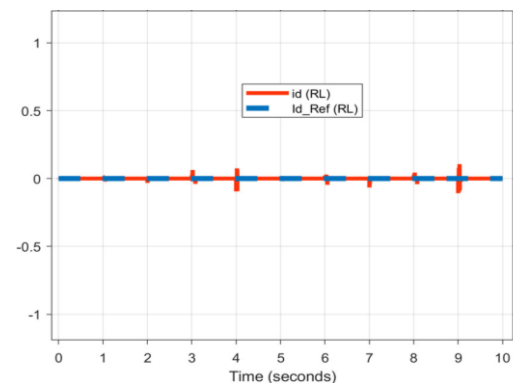


Figure 5. The i_d stator current responses of the RLAgent

Based on the results illustrated in Figure 5, the speed response of the RLAgent controller demonstrates a rapid and accurate convergence to the reference values. However, it exhibits a transient overshoot of approximately 20%. Figures 6 and 7 provide further insight into the current tracking performance, where the agent effectively follows both i_d and i_q reference signals with a steady-state error of less than 2%. The i_q current directly influences torque generation and closely matches its reference with minimal oscillation, indicating robust control behavior. Similarly, the i_d current exhibits a minor overshoot during transients but promptly stabilizes, contributing to the system's dynamic efficiency. This level of precision in current control highlights the RLAgent's capability to maintain reliable and smooth torque output for the IPMSM drive. To further assess its effectiveness, the RLAgent controller is compared to a conventional PI controller configured with $K_p=20$ and $K_i=1.07$.

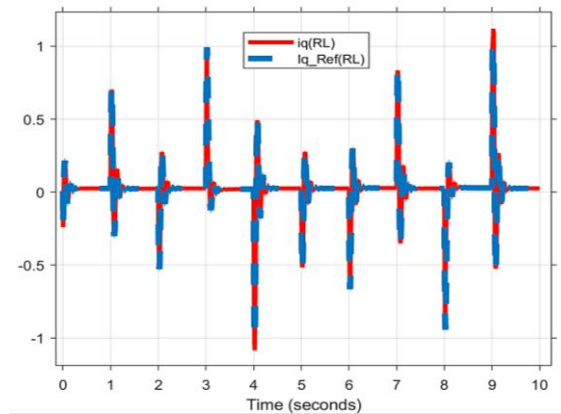


Figure 6. The i_q stator current responses of the RLAgent

As depicted in Figure 7, the RLAgent and PI controllers exhibit satisfactory speed-tracking performance during the initial 5 seconds of operation. However, the PI controller's speed response deteriorates significantly beyond this point, whereas the RLAgent maintains stable and accurate tracking. Figure 8 illustrates the d-axis current responses, where both controllers closely follow the reference signal but exhibit noticeable overshoot during the transient phase. Despite this, the RLAgent demonstrates superior recovery and overall performance. In contrast, as shown in Figure 9, the torque-generating i_q current exhibits a minor overshoot during transients but promptly stabilizes, contributing to the system's current controlled by the PI controller failing to track the reference accurately after $t=5$ seconds, leading to degraded torque output.

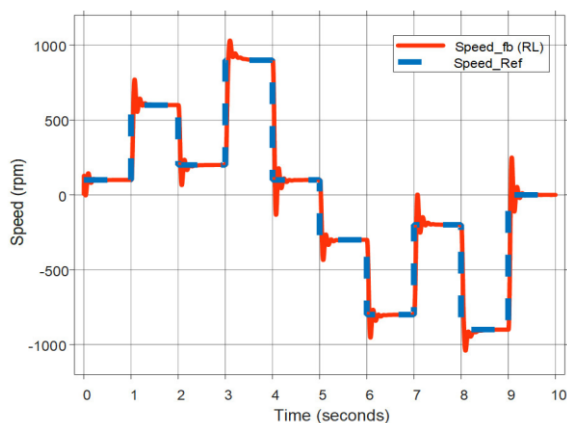


Figure 7. The speed responses of the RLAgent and PI controllers

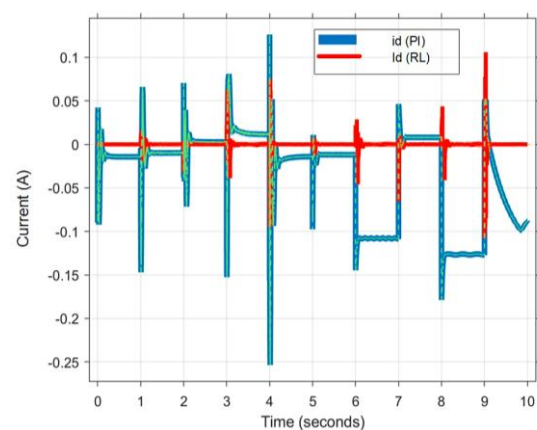


Figure 8. The i_d stator current responses of the RLAgent and PI controllers

Meanwhile, the RLAgent effectively maintains the desired i_q current and exhibits a minor overshoot during transients but promptly stabilizes, contributing to the system's dyn profile and ensuring consistent torque generation. These comparative results highlight the RLAgent's robustness and adaptability under dynamic operating conditions. While the PI controller shows acceptable performance during steady-state or low-disturbance scenarios, it lacks the precision and stability needed for high-performance applications. The RLAgent leverages learning-based adaptation to outperform traditional control methods, reaffirming the promise of RL for advanced motor control in EV systems.

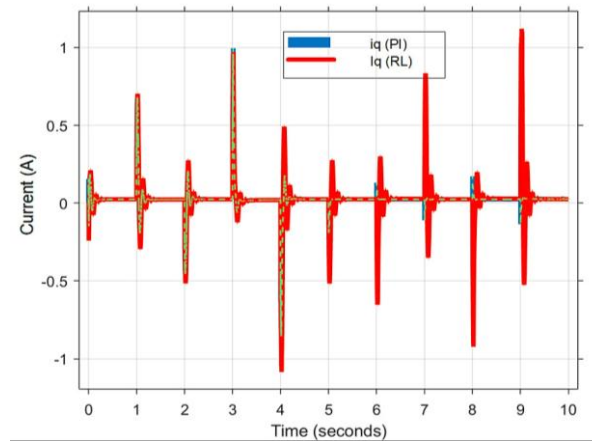


Figure 9. The i_q stator current responses of the RLAgent and PI controllers

To provide a clearer comparison between the two control methods, Table 2 summarizes key quantitative performance metrics including rise time, overshoot, steady-state error, speed tracking accuracy, and torque smoothness. These metrics highlight the superior responsiveness and precision of the proposed RLAgent controller in dynamic EV scenarios.

Table 2. Quantitative comparison of simulation results

Control method	Rise time (s)	Overshoot (%)	Steady-state error (% i_d/i_q)	Speed tracking accuracy	Torque smoothness
PI controller	0.55	24.7	~3.2%/~4.1%	Degrades after 5 s	Moderate ripple
RLAgent	0.32	19.8	~1.1%/~1.6%	Stable throughout	Smooth and accurate

As shown in Table 2, the RLAgent consistently outperforms the PI controller across all key metrics. It achieves a faster rise time (0.32 s vs. 0.55 s), lower overshoot (19.8% vs. 24.7%), and reduced steady-state errors in both i_d and i_q currents. The RLAgent maintains stable speed tracking throughout the simulation, unlike the PI controller, which degrades after 5 seconds. Additionally, the RLAgent ensures smoother torque output with minimal i_q current oscillation, enhancing drive comfort and reliability in EV applications.

5. CONCLUSION

This paper demonstrates the feasibility and effectiveness of using RL for torque control in IPMSMs for EVs. The RL agent successfully regulates d - and q - axis currents without a mathematical model, relying solely on observable system states. Simulations demonstrate that the RL-based controller outperforms conventional PI control in terms of torque accuracy, dynamic adaptability, and steady-state performance. However, the design has limitations, including speed response overshoot and sensitivity to noise during rapid load changes, as well as high computational cost and training time. Future work will focus on optimizing the reward function, expanding training for diverse conditions, and integrating adaptive noise observers to improve robustness, minimize overshoot, and enable real-time deployment in EVs.

ACKNOWLEDGMENTS

The authors thank the University of Transport and Communications for its academic support.

FUNDING INFORMATION

Vo Thanh Ha, Duong Anh Tuan, and Tran Xuan Thuy are the authors of this study, with Vo Thanh Ha serving as the corresponding and lead author. Vo Thanh Ha was primarily responsible for the research conceptualization, methodology development, formal analysis, and preparation of the original manuscript draft. Duong Anh Tuan contributed to data analysis, interpretation of results, and manuscript revision. Tran Xuan Thuy supported the study through data collection, experimental implementation, and academic proofreading. All authors have read and approved the final version of the manuscript.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Vo Thanh Ha	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	
Duong Anh Tuan	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓		
Tran Xuan Thuy	✓	✓		✓	✓	✓			✓	✓		✓		✓

C : **C**onceptualization

M : **M**ethodology

So : **S**oftware

Va : **V**alidation

Fo : **F**ormal analysis

I : **I**nterpretation

R : **R**esources

D : **D**ata Curation

O : Writing - **O**riginal Draft

E : Writing - Review & **E**diting

Vi : **V**isualization

Su : **S**upervision

P : **P**roject administration

Fu : **F**unding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest regarding the publication of this paper.

INFORMED CONSENT

We have obtained informed consent from all individuals included in this study.

ETHICAL APPROVAL

Approval was not required.

DATA AVAILABILITY

The authors confirm that the data supporting the findings of this study are available within the article and its supplementary materials.

REFERENCES

- [1] K. Zhou, M. Ai, Y. Sun, X. Wu, and R. Li, "PMSM Vector Control Strategy Based on Active Disturbance Rejection Controller," *Energies*, vol. 12, no. 20, p. 3827, Oct. 2019, doi: 10.3390/en12203827.
- [2] X. Zhang, L. Zeng, and R. Pei, "Designing and Comparison of Permanent Magnet Synchronous Reluctance Motors and Conventional Motors in Electric Vehicles," *2018 21st International Conference on Electrical Machines and Systems (ICEMS)*, Jeju, Korea (South), 2018, pp. 202-205, doi: 10.23919/ICEMS.2018.8549102.
- [3] P. T. Giang, V. T. Ha, and V. H. Phuong, "Drive Control of a Permanent Magnet Synchronous Motor Fed by a Multi-level Inverter for Electric Vehicle Application," *Engineering, Technology & Applied Science Research*, vol. 12, no. 3, pp. 8658-8666, Jun. 2022, doi: 10.48084/etasr.4935.
- [4] V. T. Ha, P. T. Giang, and V. H. Phuong, "T-Type Multi-Inverter Application for Traction Motor Control," *Engineering, Technology & Applied Science Research*, vol. 12, no. 2, pp. 8321-8327, Apr. 2022, doi: 10.48084/etasr.4776.
- [5] V. T. Ha, N. T. Lam, P. V. Tuan, and N. H. Quang, "Experiment-based Comparative Analysis of Nonlinear Speed Control Methods for Induction Motors," *Journal of Engineering and Technological Sciences*, vol. 53, no. 2, p. 210212, Apr. 2021, doi: 10.5614/j.eng.technol.sci.2021.53.2.12.
- [6] V. T. Ha and V. Q. Vinh, "Sliding mode control of a PMSM railway traction drive fed by multi-level inverter," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 21, no. 6, pp. 1405-1414, Dec. 2023, doi: 10.12928/telkomnika.v21i6.24369.
- [7] V. T. Ha and P. T. Giang, "A study on PMSM drive systems fed by multi-level inverter using linear quadratic regulator control for electric vehicle applications," *TELKOMNIKA (Telecommunication, Computing, Electronics and Control)*, vol. 21, no. 4, pp. 917-925, Aug. 2023, doi: 10.12928/telkomnika.v21i4.24432.

- [8] A. Loganayaki and R. B. Kumar, "Permanent Magnet Synchronous Motor for Electric Vehicle Applications," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Mar. 2019, pp. 1064–1069, doi: 10.1109/ICACCS.2019.8728442.
- [9] R. K. Lakhe, H. Chaoui, M. Alzayed, and S. Liu, "Universal Control of Permanent Magnet Synchronous Motors with Uncertain Dynamics," *Actuators*, vol. 10, no. 3, p. 49, Mar. 2021, doi: 10.3390/act10030049.
- [10] M. Vidlak, P. Makys, and L. Gorel, "A Novel Constant Power Factor Loop for Stable V/f Control of PMSM in Comparison against Sensorless FOC with Luenberger-Type Back-EMF Observer Verified by Experiments," *Applied Sciences*, vol. 12, no. 18, p. 9179, Sep. 2022, doi: 10.3390/app12189179.
- [11] S. S. Hakami and K.-B. Lee, "Four-Level Hysteresis-Based DTC for Torque Capability Improvement of IPMSM Fed by Three-Level NPC Inverter," *Electronics*, vol. 9, no. 10, p. 1558, Sep. 2020, doi: 10.3390/electronics9101558.
- [12] Y. Huang, J. Zhang, D. Chen, and J. Qi, "Model Reference Adaptive Control of Marine Permanent Magnet Propulsion Motor Based on Parameter Identification," *Electronics*, vol. 11, no. 7, p. 1012, Mar. 2022, doi: 10.3390/electronics11071012.
- [13] M. Nicola and C.-I. Nicola, "Improved Performance of PMSM Control Based on Nonlinear Control Law and Computational Intelligence," 2022 *International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Jul. 2022, pp. 1–8, doi: 10.1109/ICECET55527.2022.9872870.
- [14] G. Demir and R. A. Vural, "Speed Control Method Using Genetic Algorithm for Permanent Magnet Synchronous Motors," 2018 *6th International Conference on Control Engineering & Information Technology (CEIT)*, Oct. 2018, pp. 1–6, doi: 10.1109/CEIT.2018.8751896.
- [15] C.-S. Wang, C.-W. C. Guo, D.-M. Tsay, and J.-W. Perng, "PMSM Speed Control Based on Particle Swarm Optimization and Deep Deterministic Policy Gradient under Load Disturbance," *Machines*, vol. 9, no. 12, p. 343, Dec. 2021, doi: 10.3390/machines9120343.
- [16] Z. Song, J. Yang, X. Mei, T. Tao, and M. Xu, "Deep reinforcement learning for permanent magnet synchronous motor speed control systems," *Neural Computing and Applications*, vol. 33, no. 10, pp. 5409–5418, May 2021, doi: 10.1007/s00521-020-05352-1.
- [17] R. de R. Faria, B. D. O. Capron, A. R. Secchi, and M. B. de Souza, "Where Reinforcement Learning Meets Process Control: Review and Guidelines," *Processes*, vol. 10, no. 11, p. 2311, Nov. 2022, doi: 10.3390/pr10112311.
- [18] S. Singh, D. Litman, M. Kearns, and M. Walker, "Optimizing Dialogue Management with Reinforcement Learning: Experiments with the NJFun System," *Journal of Artificial Intelligence Research (JAIR)*, vol. 16, pp. 105–133, Feb. 2002, doi: 10.1613/jair.859.
- [19] Vikas, P. Yadav, B. Singh, and R. Kumar, "Model Free Reinforcement Learning based Control of Permanent Magnet Synchronous Motor Drive," 2023 *International Conference on Computer, Electronics & Electrical Engineering & their Applications (IC2E3)*, Jun. 2023, pp. 1–6, doi: 10.1109/IC2E357697.2023.10262459.
- [20] Z. Xue, Y. Wang, L. Li, and X. Wang, "An Adaptive Speed Control Method Based on Deep Reinforcement Learning for Permanent Magnet Synchronous Motor," *Proceedings of 2021 Chinese Intelligent Systems Conference*, 2022, pp. 275–286, doi: 10.1007/978-981-16-6328-4_30.
- [21] W. -L. Peng, Y. -W. Lan, S. -G. Chen, F. -J. Lin, R. -I. Chang, and J. -M. Ho, "Reinforcement Learning Control for Six-Phase Permanent Magnet Synchronous Motor Position Servo Drive," 2020 *3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, Kaohsiung, Taiwan, 2020, pp. 332–335, doi: 10.1109/ICKII50300.2020.9318882.
- [22] Y. Yu et al., "Application of Off-policy Integral Reinforcement Learning for H_∞ Input Constrained Control of Permanent Magnet Synchronous Machine," 2019 *IEEE Applied Power Electronics Conference and Exposition (APEC)*, Mar. 2019, pp. 2570–2576, doi: 10.1109/APEC.2019.8722206.
- [23] G. Eason, B. Noble, and N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 247, no. 935, pp. 529–551, Apr. 1955, doi: 10.1098/rsta.1955.0005.
- [24] A. Traue, G. Book, W. Kirchgässner, and O. Wallscheid, "Toward a Reinforcement Learning Environment Toolbox for Intelligent Electric Motor Control," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 919–928, March 2022, doi: 10.1109/TNNLS.2020.3029573.
- [25] S. Muthurajan, R. Loganathan, and R. R. Hemamalini, "Deep Reinforcement Learning Algorithm based PMSM Motor Control for Energy Management of Hybrid Electric Vehicles," *WSEAS Transactions on Power Systems*, vol. 18, no. 3, pp. 18–25, Mar. 2023, doi: 10.37394/232016.2023.18.3.

BIOGRAPHIES OF AUTHORS



Vo Thanh Ha    received a B.S. in Control and Automation Engineering from Thai Nguyen University of Technology in 2002, and a master's degree from Hanoi University of Science and Technology in 2004. She completed her Ph.D. in 2020 at the same institution. She has been a lecturer at the University of Transport and Communications since 2005. Her research focuses on electrical drives, robot control, electric vehicles, and power electronics. She has authored and co-authored several papers published in Web of Science-indexed journals (SCIE/ESCI). She can be contacted at email: vothanhha.ktd@utc.edu.vn.



Duong Anh Tuan    received a B.S., M.S., and Ph.D. degrees from Hanoi University of Science and Technology, Vietnam, in 2002, 2007, and 2023, respectively, all in Control Engineering and Automation. Since 2004, he has been employed at Hanoi University of Industry, where he is a lecturer and researcher at the Faculty of Electrical Engineering. His research interests include modeling and controlling of multi-level inverter for applications such as photovoltaic, wind systems, and electric vehicles. He can be contacted at email: tuanda@hau.edu.vn.



Tran Thuy Van    is currently a faculty member in the Department of Industrial Electricity, Faculty of Electrical Engineering, Hanoi University of Industry, Hanoi, Vietnam. He received his B.S. degree in Industrial Measurement and Computing from Hanoi University of Science and Technology in 2003, followed by an M.S. degree in Measurement and Control Systems from the same university in 2006. He earned his Ph.D. in Control Science and Engineering from Hunan University, China, in 2016. His research interests include measurement engineering, adaptive and optimal control, robotics, and artificial intelligence. He can be contacted at email: vanntt@hau.edu.vn.